International Journal of Pure and Applied Mathematics
Volume 119 No. 14 2018, 1571-1582
ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue
ijpam.eu

AN EXPERIMENTAL STUDY OF DYNAMIC CONTROLLER ASSIGNMENT PROBLEM IN SDN NETWORKS

Pragyan ashok masalka1[1], Dr.Himanshu Agrawal[2]
[1,2]CS-IT, Dept. SIT, Symbiosis International University, Pune, India.

ABSTRACT -- The assignment of routers and switches to servers and other controllers can be viewed as a static and dynamic assignment problem. Static controller assignment problem has been studied a lot by various researchers. To address the static controller assignment problem, majority of the solutions are based on Hungarian algorithm. Modern data networks are highly distributed and hence assignment of resources can be viewed as dynamic assignment problem, wherein various networking resources such as switches and routers should be assigned dynamically (i.e. on run time) to the controller to handle the network load. Therefore, in the past few years, there is growing interest to use Software Defined Networking (SDN). As opposed to the traditional network using the same data and control plane, SDN decouples the data and control plane to address the distribution of data and assignment of routers and switches to the SDN controller dynamically. The paper studies a dynamic controller assignment problem and proposes an Adaptive Controller. Assignment using Hungarian and genetic (ACAHGA).To study the dynamic assignment problem, we conducted an experiment using a small test-bed of four machines. The performance of the proposed algorithm is measured in terms of latency and load balancing.

Keywords — SDN, RFHC, Data center, Adaptive Hungarian, genetic

INTRODUCTION

Modern data networks are highly distributed and hence assignment of resources can be viewed as dynamic assignment problem, wherein various networking resources such as switches and routers should be assigned dynamically (i.e. on run time) to the controller to handle the network load [1]. In the past few years, there is growing interest to use Software Defined Networking (SDN) for the dynamic controller assignment. As opposed to the traditional network using the same data and control plane, SDN decouples the data and control plane to address the distribution of data and assignment of routers and switches to the SDN controller dynamically [2].

As depicted in Figure.1a, the traditional network comprises switches, operating and packet forwarding hardware. There are range of protocols, functions being performed by the routers, firewall, and switches in the traditional network such as Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), Network Address Translation (NAT), traffic engineering, and differentiated services. When the network conditions changes dynamically, it becomes really challenging for the traditional network to efficiently manage the network load and resources, therefore resulting in network congestion and increased delay problem for various services [10]. Moreover, the assignment of the switches and routers is static. Hence any change in the network condition is exhibited by new control requirements, thus the traditional network fails to handle such dynamic traffic scenario and is able to provide a very limited local view of the network while managing traffic On the other hand, SDN based network uses Network Operating System, a SDN controller and thus provides a global view of the network thereby enabling dynamic assignment of switches to the controller. SDN uses open source protocols such as Open Flow [37] to accept the input from Internet Service Provider (ISP). As shown in figure 1b, a SDN based network comprises of control program and Network Operating System (NOS), thus offers global view of network. It becomes easy to write, maintain, and verify network functions using NOS.

The paper is motivated by the recent developments on dynamic controller assignment problem (DACP) in the SDN enabled data centres [1]**.** The DCAP is formulated as an optimisation problem to handle the tasks assignment efficiently. Task assignment or resource assignment problem is a classical problem and has been studied by many researchers using Hungarian algorithms [22, 7].

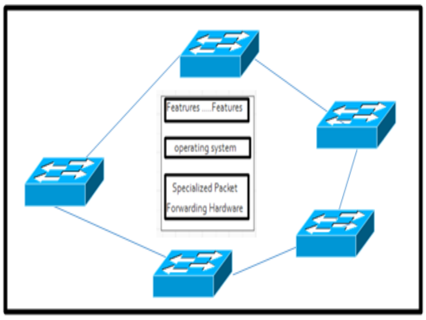

Figure 1a: The Traditional network

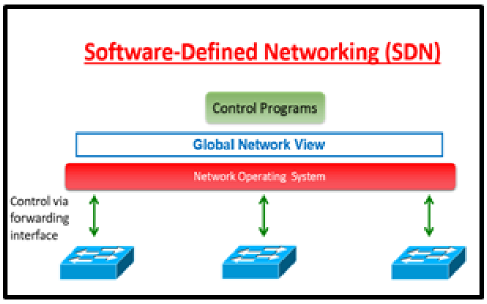

Figure 1b: A SDN based network

Our contribution in this paper is as follows:

1. We conducted an extensive literature review on the dynamic controller assignment problem in the context of SDN enabled data center networks. Application of DCAP for SDN enabled network is relatively new and is an open area of research.

2. We conducted an experimental study to analyses the performance of the proposed Adaptive Controller Assignment Hungarian Genetic Algorithm. Results shows that the proposed algorithm provides less latency and improved throughput as compared to the two previous algorithms.

The notations are mentioned in the table no 1, which are extensively used in the paper for better narration.

Definitions and notations

Notation	Denoted as
f_{set}	Set of folders
C_{set}	Cluster set
T_{set}	Time set
D_{set}	Data set
Cr	Controller thread
C_l	Controller
f_t	Fitness time
A_{cd}	Assignment control data
O_{al}	Optimal assignment list
T_w	Time window
S_c	Server score
E_T	Expected time
T_B	Total byte
T	Knock time
K_B	Knock byte
OPT	Optimal scheme
MM	Master assignment
LCMA	Low complexity master assignment
ms	Milli second

1. RELATED WORK

The dynamic assignment problem has been studied extensively by various researchers in the last few years. This section presents a review on dynamic controller Assignment problem in Software Defined Networks.
Author of [2]did the study on assignment problems where in [3] gave idea of allocation of resources in shared computer systems and [4] made the formulation for dynamic assignment problem and in [5] researcher tried to solved dynamic traffic assignment method for planning and telematics applications author of [6] successfully solved dynamic assignment problems in [7] dynamic Hungarian was used to solved assignment problem and [8] used for the comparison of static traffic assignment problem with dynamic traffic assignment problem is done.[9]experiment done on dynamic flow in data center networks.[10] improving energy efficiency via multi-controller SDN in data center network [11]online algorithm is proposed for geographical load balancing [12]an online cost minimizing is done by moving big data to the cloud.[13] encryption performed in networking using RCC to provide network security.[14]globally deployed software define WAN experienced in B4.[15] dynamic controller provisioning in SDN is done with basic initial state and reassignment algorithm.[16]elastically SDN controllers are distributed[17]efficiently elastic distribution of SDN in control plane [18]through simulation multipath routing is done in SDN for fat tree data center networks[19]started doing deep packet inspection to detect intrusion[20]for usage of less memory data compression technique is used over data in network[21]multiple task allocation takes place through assignment algorithms[22]traffic is being controlled

with the help of virtualized network function in SDN enabled data centers[23]load balancing of energy efficient in SDN based data center networks[24]for application aware SDN enables in data center networks[25]stable matching algorithm is being used to have control over transferring the data[26]virtually paths are assigned for assignments to maintain load-balancing.[27]SDN based channels are maintained with the help of dense Wi-Fi network.[28]study of different switching latency is being taken place for SDN based routing in multihop,multiradio,wireless mesh networks[29]how to synchronize data when flooding attacks in SDN based networks taken place[30]For online social networking services the bulk of data is pushed using heterogeneous wireless networks[31]to achieve reliability in SDN path control management is done.[32]for inter-data center SDN allows to perform multipath forwarding[33]due to SDN based network its become easy to overcome the memory limit.[34]multipath with TCP and segment routing in SDN based data center networks[35]heavy data storage capacity is being checked through information centric networks in SDN data center network[36]virtually load balancing is being done in cloud data centers using SDN[37] first by knowing the energy in packets of data than routing decision taken place in data center networks with SDN[38]unified programmability is designed for VNF and SD wireless networks.[39]5G mobile core networks is based on SDN and NVF[40]to maximize the throughput an efficient algorithms are used in SDN with consolidated middle box.[41]improving robustness in SDN to Avoid attack in network[42]dynamic flow scheduling for optical data centers in SDN.[43]for fast restoration in SDN the different optimization methods are used.[44]flow monitoring is done in SDN to reduce congestion and packet loss[46]for large scale SDN the decentralized monitoring is being done.[47]a light weight solution is provided for data plane state recovery in SDN [49]stateful monitoring on SDN is done to protect the network from DDOS.[50]data shuffling is done in files to make it more secure during routing[51]evaluating the DOS attack in SDN to know the impact of DOS on network[52]load balancing for multiple incoming resources 3through virtual network functions[53]detecting and mitigation DOS attacks over the data plane in SDN.[1]An efficient online algorithm for dynamic SDN controller assignment in data center networks is done with multiple concept where, stable matching phase procedure output is given as input to coalitional game phase procedure.DCAP online and offline algorithms were proposed for approximation. The RFHC is used with DCAP to decompose the time series. Reduce cost and iteration and to improve overall performance with their proposed algorithms the whole process is being carried out in simulation.

To overcome the different simulation techniques being used till date we have formed the live dummy network to show the working of SDN for assigning the different task to different controller server where throughput, latency and overall performance improves using DCAP Hungarian genetic algorithm due to this system started doing the self-learn and gives the best fit value for network.

2. PROPOSED METHODOLOGY

Problem statement

Given the number of active SDN controllers and their processing capacity. How to find an optimal dynamic controller assignment in data center network scenario can be view in equation 1.

$$f(\text{Dac}) = \sum_{i=1}^{m} \quad \sum_{j=1}^{n} \quad \Rightarrow \text{Ga(Ta)}\text{-----------}(1)$$

Where,

f(Dac)=function of dynamic assignment controller
m=number of controllers

n=number of task processing threads

Ga(Ta)=task assignment through genetic algorithm
Hence, find through the genetic algorithm and assignment is through improved Hungarian algorithm.

System overview and working

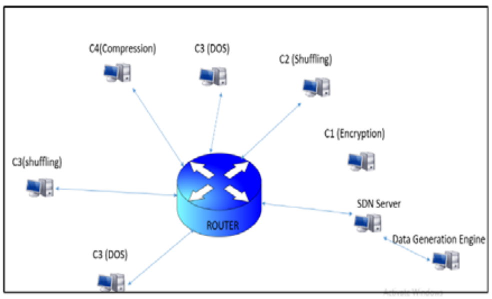

Fig2: system overview

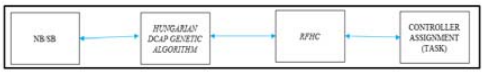

Fig3: Flow inside SDN server

The proposed methodology of controller assignment problem can be explained with below mentioned steps:
Step 1: This is the first step where the data is given as input which contains all format of files with different file size. This is been taken by system to form 2-D vector which contain file name and its size. Collection of data at assignment server is shown as below in equation 2:

$$f\ (\text{DAsn}) = \sum_{i=1}^{n} Dv \text{ --------------(2)}$$

Where, 1<i<n

Where,

$f(\text{DAsn})$=function for data collection at assignment server

n=number of received data

Dv =data vector sets

That can be shown as,

Dv = {D1, D2, D3, D4…………………..DN}

Step 2: MPTCP - This is a step where using UDP and TCP protocol the assignment server knocks the controllers with labelled message to record the time delay. This eventually helps the proposed model to design randomized horizon control algorithm.

Step 3: North Bound and South Bound - The most of the assignment problem is considered to solve the traffic congestion and thereby to provide a smooth assignment of data to the controllers within the data center. So many methods they do deal with mice and elephant data to take right routing decision with in data centres. But this is eventually creates a hurdle for other size of data.

So, this step consider all size of data to cluster them with ranges like low, very low,medium,high,very high using optimized fuzzy-c Means clustering technique as depicted in algorithm.

Unlike Fuzzy c means clustering, which generally takes Apriori specifications of clusters which increase the iteration cost of the methodology. To improve this proposed algorithm incorporates the Fuzzy C means clustering technique uses the five fuzzy crisp values to evaluate the underlying factors of clustering.

ALGORITHM 1: FCM

//**INPUT**: Let D= {f_1, f_2, f_3 ..., f_n} be set of data folder and

//**OUTPUT**: CLUSTER SET C_{set} = {c_1, c_2, c_3 ..., c_c}

1) **Step 1:** start
2) T_{set}=Φ, D_{Tset}=Φ, C_{set}=Φ
3) ***for*** i=0 to size of D
4) $T_{set[0]}$=$D_{i\ name}$, $T_{set[1]}$=$D_{i\ size}$
5) add $T_{set[0]}$ to D_{Tset}
6) ***end for***
7) Set big = $D_{Tset[0]}$.$T_{set[1]}$
8) Small = $D_{Tset[0]}$. $T_{set[1]}$
9) ***for*** i=0 to size of D_{Tset}
10) ***If***(big< $D_{Tset[i]}$.$T_{set[1]}$)
Big= If (big< $D_{Tset[i]}$.$T_{set[1]}$);
11) ***If***(small< $D_{Tset[i]}$.$T_{set[1]}$)
small= If (big< $D_{Tset[i]}$. $T_{set[1]}$);
12) ***end for***
13) D=(big-small)/5
14) C_{Rset}=Φ [fuzzy crisp set]
15) ***for*** i=1 to 5
16) add d to C_{set};
17) ***end for;***
18) ***for*** i=0 to size of C_{Rset}
19) C_R=C_{Rset} [cluster range]
20) ***for*** j=1 to size of D_{Tset}
21) ***If***($D_{Tset[i]}$. $T_{set[i]}$ ∈ C_R
22) add $D_{Tset[i]}$.$T_{set[i]}$ to C_{set}
23) ***end for***
24) ***end for***
25) Return C_{set}
26) Stop

This process of NB- SB can be depicted with the below mentioned equation 3

$$f(\text{NbSb}) = \int_{0}^{min} \int_{min}^{max} Dv \text{ ------------(3)}$$

$$Dv \Rightarrow \sum_{k=1}^{n} Dc$$

Where, 1<k<n

Where,

$f(\text{NbSb})$=function for north bound and south bound

n =number of clusters

Dc=cluster sets

Step 4: Hungarian DCAP Genetic Algorithm-Here in this step the data clusters are formulated to transfer in network through the server based on improved Hungarian task assignment algorithm. Which is enhanced by using genetic algorithm? The Steps which are taken to achieve this is explained as below.

Initial Population:

Individual Controllers which are formed in the previous step is loaded into a list to estimate the performance time by the different threads. so proposed system forms same number of threads dynamically as of number of Controllers.

So this step yields a matrix of M X N where rows (M) indicate number of Controllers and column (N) indicates the number of threads. And this constitutes the initial population of the genetic algorithm where each entity of the matrix is the performance time in milliseconds of the threads for the respective controllers.

Fitness Function:

Here this step estimates the time taken for the given population matrix of M X N. Time is estimated based on the Hungarian process of task allocation which works on the principle of one task at an instance for one thread based on the fact of working strategy of processor.

In this process wait time and performance times are been evaluated for the given controllers execution and total time is estimated as the fitness function for the instance generation and it is assumed to be smallest.

Selection:

This is the step where best generation is retained based on the selection of low time parameter with the past generation fitness function.

If the current generation is having higher value of fitness function then its selection is discarded and the past generation is retained for the next process.

Whereas if the current generation is having lesser fitness function value compared to past one, then it is selected for the next process and the past generation population will be discarded.

Crossover:

This is the most significant important phase of the genetic algorithm where instance generation matrix is get regenerated. This is done by combining the positions of the controllers to generate new combination using permutation of controllers with respect to the performance threads.

This is done by estimating the permutation of the controllers given by the equation 4 as follows

Number of all permutations of *n* controllers, taken *r* at a time, is given by:

$^{n}P_{r} = n(n - 1)(n - 2) \ldots (n - r + 1) = n!/ (n - r)!$ ------ (4)

Mutation:

This is the step where newly formed offspring (that means in our case newly estimated time through the formed cross over matrix) is checked for the number of fixed generation to terminate the process. This can be made either by nominal changes in the offspring combination or it can be retain as it is, if it is yields satisfactory results. The whole process of Hungarian genetic algorithm will be depicted in algorithm 1.

DCAP: Here the controllers are selected based on the past assignment weight that is extracted through the time congestion in the network. Here the controllers which are yielding more time are discarded and best performed controllers are selected dynamically with the help of Genetic Algorithm.

Here in this step the obtained combination of controllers with respect to the threads through Hungarian genetic algorithm are loaded to the threads in the same manner. The Algorithm for *Hungarian* DCAP Genetic Algorithm is depicted in algorithm 2.

Hungarian method generally uses multi graph factor to get the assignment task done, this eventually takes larger number of iterations to get the matched vertices from the graph. So proposed model improves this by deploying genetic algorithm along with the Hungarian basic principles for controlled number of generations with high mutation level. This is catalyzed by using DCAP which eventually chooses the best generation based on the past performance of assignment problem.

Algorithm 2: HUNGARIAN DCAP GENETIC ALGORITHM

Input: No. of controllersC_r
No. of data Threads C_l
Time set = ($c_{r1}c_{l1}$, cr2cl2, crn, cln)
Output: An optimal assignment listO_A

1) Start
2) Set gen_size,gen=n,i=0,A_{SL}=null
3) ***While***(i≤gen)
4) ${}^{n}p_r$ C_l;

Where [${}^{n}p_r$ =permutation of C_l]

5) ***If*** i==0
6) T_c,T_p=gettotal _time (C_r, C_l, T_{set})
7) ***else***
8) T_c=gettotal _time (C_r,C_l,T_{set})
9) ***If***(T_c<T_p)
10) f=T_c

Where, f=fitness function

11) restore gen g_c
12) ***else***
13) f_t=T_p
14) restore gen g_p
15) ADD g_p to A_{SL}
16) discard g_c
17) i++;
18) ***end while***
19) ***return*** A_{SL}

ALGORITHM 2A: **Time evolution method**
(Method called in Improved Hungarian with genetic algorithm)

//**INPUT**: No. of controllers C_r No. of data threads C_lTimeset T_{set} = (C_{r1} C_{l1}, C_{r2} C_{l2}, C_{r3} C_{l3}….. C_{rn} C_{ln})

//**OUTPUT:** total time
Function: gettotal _time (C_r, C_l, T_{set})

1) Start
2) total time =0
3) ***for*** i=0 to size of C_l
4) ***for*** j=0 to size of C_r
5) ***if*** j=0 then
6) ***if*** C_{rj} is engaged
7) $T_{C_{rj}C_{li}}$=$T_{C_{rj}C_{li}}$+$T_{set}C_{rj}C_{li}$

Where [T: Time]

8) ***else***
9) ***if*** C_{rj} is engaged
10) ***if*** $T_{pC_{rj}C_l}$>$T_{cC_{rj}cC_{li}}$
11) $T_{C_{rj}C_{li}}$ =$T_{pC_{rj}C_l}$+W_t+$T_{set}C_{rj}C_{li}$

Where [W_t = *waiting time*]

12) total time=total time+$T_{C_{rj}C_{li}}$
13) end ***for***
14) end ***for***
15) ***return*** total time
16) stop

Improved Hungarian representation in equation form in equation (5):

$$f(\text{Ih}) = \sum_{i=1}^{m} \sum_{j=1}^{n} Dc \Rightarrow \sum_{h=1}^{x} \sum_{p=1}^{y} Ta \text{---(5)}$$

Where,

f(Ih)=function of Improved Hungarian
m=number of controllers
n=number of data threads
p=number of pattern
Dc =instant data list from improved Hungarian assign matrix will be formed
x=number of assigned task to controllers
y=number of data
Ta =Task assignment function

Step 5: Stable matching and RFHC –in this step a time window will set to measure a stable matching score using RFHC algorithm, a ratio is being calculated using the knock time and total bytes for the given time window for a specific controller and then based on this a reassignment and optimization of process is done to assign proper controllers there by to stable the whole process of controller assignment with high accuracy and whole process is depicted in algorithm 3.

Unlike the basic RFHC which always chooses random integer to fix the time window, proposed model uses the time response factor of each controller to fix the time window. This may optimizes the assignment of the specific controllers.

ALGORITHM 3: RFHC

//**INPUT**: A_{CD} [assignment control data]

//**OUTPUT**: optimal assignment list [O_{AL}]

1) start
2) set time $T_{w;}$ [Time window]
3) set S_c =0 [server score]
4) set E_T=(T_B *T)/K_B [E_T : expected time, T_B : total byte, T :knock time, K_B :knock byte)
5) i=0, T_s=0
6) while (A_{CD} is not empty)
7) T_{set} = $A_{CD\,i}$
8) Size = $T_{set[1]}$
9) T_S=T_S+ size
10) ***If***(T_w (E_T)>E_T)
11) S_{c++}
12) add s_c O_{AL}
13) ***end if***
14) ***end while***
15) Return O_{AL}
16) Stop

3. RESULT AND DISCUSSION

Proposed model of ACA-HG is deployed in real time live server and controllers. For the experimental evaluation proposed model uses 4 laptops. Each are having average CPU of core i3 with primary memory of 4GB. Each of the machine is powered with windows operating system and all are java enabled. For the development of the model system uses Netbeans 8.0 as IDE. Proposed system uses D-Link as wireless router to establish a wireless LAN.

Response Time Evaluation:.

When the proposed model ACA-HG which is incorporated in live scenario is compared with the [1], which runs in simulation setup. Our model yields better transfer rate based on the response time, this is due to limited iterations carried out by the Hungarian genetic algorithm that are using to identify best controller with minimum traffic overheads.

[1] Uses the stable matching transfer protocol which is performing moderate number of iterations to perform the task. SMT(Stable matching transfer) model yields average controller response time of 0.01836 seconds and ACA-HG yields average controller response time as 0.01152 seconds. This clearly indicates the ACA-HG yields better response time than the SMT model.

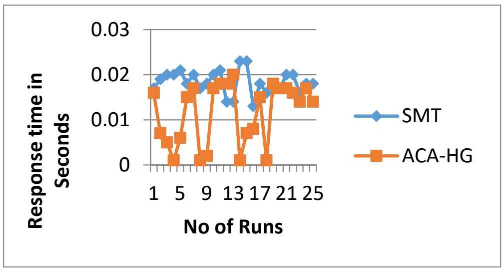

Figure 4: Comparison between SMT and ACA-HG

Latency Evaluation:

Latency in network can be defined as the delay between the sender and receiver nodes. When the latency of the proposed model of ACA- HG is compared with that of [54] some facts are revealed. [54] basically works on the principle of LCMA (low-complexity master assignment)that is low cost master assignment algorithm, which works on the basis of the average flow control technique over the period of assignment to the controllers this take considerable more time for the given span.

Whereas the proposed system works on the Hungarian genetic algorithm which is catalyzed with the RFHC algorithm to decide the assignments over the given instance. So the performance of the system is high compared to that of LCMA method. This can be shown in the below table 1 and plot 2.

OPT(optimal scheme) and (MM)master assignment scheme

Table 2: Latency of different techniques

Methods	Latency In ms
OPT	6.45
LCMA	6.45
MM	7.6
ACA-HG	4.59

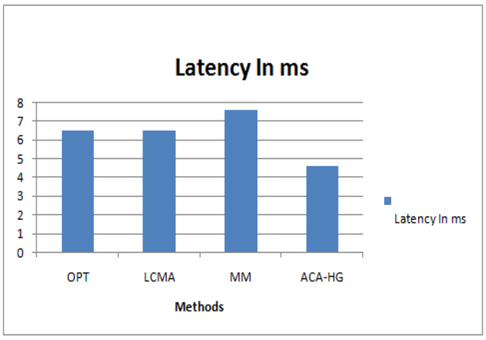

Figure 5: Latency Comparison between other techniques.

Throughput:

Throughput can be defined as the amount of data transferred in unit time over the network. Proposed model measures the throughput for different size of the data to evaluate the flexibility of the system. The Recorded throughput for different experiment is shown below

Table 2: Throughput evaluation

Total Data Size in MB	Trasnferred in MS	Throughput KBPS
4.52	11563	390.902015
10.56	14878	913.2578051
22.56	22774	1951.050765
40.58	31002	3509.469861
80.77	44867	6985.21145

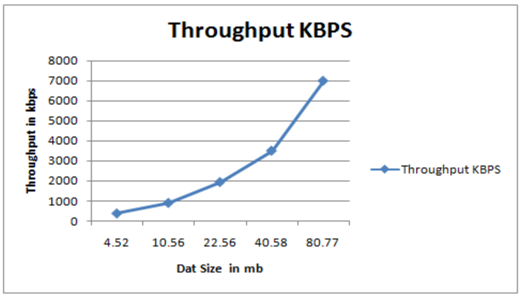

Figure 6: Throughput Evaluation

On observing the above graph we come to know that proposed model throughput become better and better as the data increases. This can be seen in the above plot as the warm steep rather than stand in 45 degree. This is because of the quick assignment of the controller in limited threshold of iterations.

The Proposed model of controller assignment in software defined network using improved Hungarian and Genetic algorithm achieves higher response time in between the server and the controller compare to that of [1]. This is mainly due to proposed model uses MP- TCP as routing protocol, which eventually associates with the UDP and the FTP protocols in our real time deployment of the model. As we know that in real time network paradigm of a healthier state, average bandwidth for the data transmission is about 100 MBPS. This boosts our response time, which is even more improved due to fast controller assignment in the network by our model.

Our proposed model uses very constrained environment to evaluate the assignment problem using limited iterations in Genetic algorithm. The genetic algorithm of the proposed model finely blended with the improved Hungarian algorithm, where it uses matrix transition method to evaluate the offspring of the generations, which eventually produces the best assignment list for the controllers. Hungarian algorithm of [21] is used based on the bipartite graph which takes more time to traverse over the edges and nodes, Where as our model improves this using matrix transition process.

This assignment lists from the Hungarian genetic algorithm is yielded due to selection and discard process of the offspring by the DCAP algorithm. Whereas [1] uses the stable matching algorithm which is basically developed in simulation environment. The stable matching algorithm uses moderate amount of iterations for the controller assignment which is slightly higher than that of our model.

These all the processes of the proposed model is the reason for why our model yields good response time than that of [1].

Proposed model uses the RFHC algorithm to enhance the instance data flow rate which is learned from the past runs. Due to this proposed model achieves good latency compared to that of [54],. Where [54] uses the average flow control technique over the period of assignment to the controllers this take considerable more time for the given span.
By analyzing all the facts we could have easily say that our model achieves good throughput. Which is tabulated with some experimental runs as mentioned in this section.

4. CONCLUSION

Unlike many methodologies on SDN our proposed model of ACA-HG not deploys the system in simulation environment. As a change and challenge proposed model chooses to incorporate the controller assignment problem in real time environment with 4 number of controllers and one assignment Server. Proposed model achieves better response time than that of simulation environment, this is due to usage of strong routing protocols of MP/TCP in the controlled environment. Again ACA-HG achieves better latency as the model uses of limited iterations for assignment problem using Hungarian and genetic algorithm, which is also having feature of adaptiveness in the assignment problem by deploying RFHC algorithm.

As future enhancement of the system, the model can be incorporate in real time cloud controllers to handle big data in complex environments.

REFERENCES

[1] Wang, T., Liu, F., & Xu, H. (2017). An Efficient Online Algorithm for Dynamic SDN Controller Assignment in Data Center Networks. IEEE/ACM Transactions on Networking, 25(5), 2788-2801.

[2] Gale, D., & Shapley, L. S. (1962). College admissions and the stability of marriage. The American Mathematical Monthly, 69(1), 9-15.

[3] Jain, R., Chiu, D. M., & Hawe, W. R. (1984). A quantitative measure of fairness and discrimination for resource allocation in shared computer system (Vol. 38). Hudson, MA: Eastern Research Laboratory, Digital Equipment Corporation.

[4] Powell, W. B. (1996). A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. Transportation Science, 30(3), 195-219.

[5] Friedrich, M., Hofsaß, I., Nokel, K., & Vortisch, P. (2000). A dynamic traffic assignment method for planning and telematic applications. PTRC-PUBLICATIONS-P, 29-40.

[6] Kogan, K., Khmelnitsky, E., & Ibaraki, T. (2005). Dynamic generalized assignment problems with stochastic demands and multiple agent--task relationships. Journal of Global Optimization, 31(1), 17-43.

[7] Mills-Tettey, G. A., Stentz, A., & Dias, M. B. (2007). The dynamic hungarian algorithm for the assignment problem with changing costs.

[8] Eleuterio, V. D. S., & Zurich, I. E. (2007). Static Traffic Assignment Problem. A comparison between Beckmann (1956) and Nesterov & de Palma (1998) models.

[9] Al-Fares, M., Radhakrishnan, S., Raghavan, B., Huang, N., & Vahdat, A. (2010, April). Hedera: Dynamic flow scheduling for data center networks. In Nsdi (Vol. 10, pp. 19-19).

[10] Zhang, P., & Hu, D. (2010). E 3 MC: Improving Energy Efficiency via Elastic Multi-Controller SDN in Data Center Networks. Power, 2011, 2012.

[11] Lin, M., Liu, Z., Wierman, A., & Andrew, L. L. (2012, June). Online algorithms for geographical load balancing. In Green Computing Conference (IGCC), 2012 International (pp. 1-10). IEEE.

[12] Zhang, L., Wu, C., Li, Z., Guo, C., Chen, M., & Lau, F. C. (2013). Moving big data to the cloud: An online cost-minimizing approach. IEEE Journal on Selected Areas in Communications, 31(12), 2710-2721.

[13] Isaac, E. R., Isaac, J. H., & Visumathi, J. (2013, February). Reverse Circle Cipher for personal and network security. In Information Communication and Embedded Systems (ICICES), 2013 International Conference on (pp. 346-351). IEEE.

[14] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., ... & Zolla, J. (2013, August). B4: Experience with a globally-deployed software defined WAN. In ACM SIGCOMM Computer Communication Review (Vol. 43, No. 4, pp. 3-14). ACM.

[15] Bari, M. F., Roy, A. R., Chowdhury, S. R., Zhang, Q., Zhani, M. F., Ahmed, R., & Boutaba, R. (2013, October). Dynamic controller provisioning in software defined networks. In Network and Service Management (CNSM), 2013 9th International Conference on (pp. 18-25). IEEE.

[16] Dixit, A., Hao, F., Mukherjee, S., Lakshman, T. V., & Kompella, R. (2013, August). Towards an elastic distributed SDN controller. In ACM SIGCOMM Computer Communication Review (Vol. 43, No. 4, pp. 7-12). ACM.

[17] Krishnamurthy, A., Chandrabose, S. P., & Gember-Jacobson, A. (2014, August). Pratyaastha: an efficient elastic distributed sdn control plane. In Proceedings of the third workshop on Hot topics in software defined networking (pp. 133-138). ACM.

[18] Jo, E., Pan, D., Liu, J., & Butler, L. (2014, December). A simulation and emulation study of SDN-based multipath routing for fat-tree data center networks. In Simulation Conference (WSC), 2014 Winter (pp. 3072-3083). IEEE.

[19] Parvat, T. J., & Chandra, P. (2014, December). Performance improvement of deep packet

inspection for Intrusion Detection. In Wireless Computing and Networking (GCWCN), 2014 IEEE Global Conference on (pp. 224-228). IEEE.
[20] Immareddy, S., & Talusani, S. K. (2014, December). Multi level huffman test data compression with multiple scan chain method. In Computer and Communications Technologies (ICCCT), 2014 International Conference on (pp. 1-6). IEEE.
[21] Nam, C., & Shell, D. A. (2015). Assignment algorithms for modeling resource contention in multirobot task allocation. IEEE Transactions on Automation Science and Engineering, 12(3), 889-900.
[22] Leivadeas, A., Falkner, M., Lambadaris, I., & Kesidis, G. (2016, October). Dynamic traffic steering of multi-tenant virtualized network functions in sdn enabled data centers. In Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), 2016 IEEE 21st International Workshop on (pp. 65-70). IEEE.
[23] Carlinet, Y., & Perrot, N. (2016, September). Energy-efficient load balancing in a SDN-based Data-Center network. In Telecommunications Network Strategy and Planning Symposium (Networks), 2016 17th International (pp. 138-143). IEEE.
[24] Aziz, M., Fazely, H. A., Landi, G., Gallico, D., Christodoulopoulos, K., & Wieder, P. (2016, December). SDN-enabled application-aware networking for data center networks. In Electronics, Circuits and Systems (ICECS), 2016 IEEE International Conference on (pp. 372-375). IEEE.
[25] Wang, T., Liu, F., Guo, J., & Xu, H. (2016, April). Dynamic SDN controller assignment in data center networks: Stable matching with transfers. In INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE (pp. 1-9). IEEE.
[26] Wang, X., Huang, C., He, K., Fan, X., & Fu, B. (2016, December). Virtual Path Assignment Based on Load Balancing for SDNs. In Parallel and Distributed Systems (ICPADS), 2016 IEEE 22nd International Conference on (pp. 247-255). IEEE.
[27] Seyedebrahimi, M., Bouhafs, F., Raschellà, A., Mackay, M., & Shi, Q. (2016, June). SDN-based channel assignment algorithm for interference management in dense Wi-Fi networks. In Networks and Communications (EuCNC), 2016 European Conference on (pp. 128-132). IEEE.
[28] Sajjadi, D., Tanha, M., & Pan, J. (2016, January). A comparative study of channel switching latency for conventional and SDN-based routing in multi-hop multi-radio Wireless Mesh Networks. In Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual (pp. 330-334). IEEE.
[29] Mohammadi, R., Javidan, R., & Conti, M. (2017). SLICOTS: an SDN-based lightweight countermeasure for TCP SYN flooding attacks. IEEE Transactions on Network and Service Management, 14(2), 487-497.
[30] Ko, H., Lee, J., & Pack, S. (2017). An Opportunistic Push Scheme for Online Social Networking Services in Heterogeneous Wireless Networks. IEEE Transactions on Network and Service Management, 14(2), 416-428.
[31] Song, S., Park, H., Choi, B. Y., Choi, T., & Zhu, H. (2017). Control path management framework for enhancing software-defined network (SDN) reliability. IEEE Transactions on Network and Service Management, 14(2), 302-316.
[32] Wang, Y. C., Lin, Y. D., & Chang, G. Y. (2017, June). SDN-based dynamic multipath forwarding for inter-data center networking. In Local and Metropolitan Area Networks (LANMAN), 2017 IEEE International Symposium on (pp. 1-3). IEEE.
[33] Marsico, A., Doriguzzi-Corin, R., & Siracusa, D. (2017, May). Overcoming the memory limits of network devices in SDN-enabled data centers. In Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on (pp. 897-898). IEEE.
[34] Pang, J., Xu, G., & Fu, X. (2017). SDN-Based Data Center Networking With Collaboration of Multipath TCP and Segment Routing. IEEE Access, 5, 9764-9773.
[35] Atiyah, M. H., & Jasim, A. D. (2017, March). Supporting Information-Centric Networking in SDN data center network. In New Trends in Information & Communications Technology Applications (NTICT), 2017 Annual Conference on (pp. 69-74). IEEE.
[36] Tsygankov, M., & Chen, C. (2017, June). Network aware VM load balancing in cloud data centers using SDN. In Local and Metropolitan Area Networks (LANMAN), 2017 IEEE International Symposium on (pp. 1-6). IEEE.
[37] Ghuman, K. S., & Nayak, A. (2017, May). Per-packet based energy aware segment routing approach for Data Center Networks with SDN. In Telecommunications (ICT), 2017 24th International Conference on (pp. 1-6). IEEE.
[38] Schulz-Zander, J., Mayer, C., Ciobotaru, B., Lisicki, R., Schmid, S., & Feldmann, A. (2017).

Unified Programmability of Virtualized Network Functions and Software-Defined Wireless Networks. IEEE Transactions on Network and Service Management, 14(4), 1046-1060.

[39] Basta, A., Blenk, A., Hoffmann, K., Morper, H. J., Hoffmann, M., & Kellerer, W. (2017). Towards a cost optimal design for a 5G mobile core network based on SDN and NFV. IEEE Transactions on Network and Service Management, 14(4), 1061-1075.

[40] Huang, M., Liang, W., Xu, Z., & Guo, S. (2017). Efficient Algorithms for Throughput Maximization in Software-Defined Networks With Consolidated Middleboxes. IEEE Transactions on Network and Service Management, 14(3), 631-645.

[41] Rueda, D. F., Calle, E., & Marzo, J. L. (2017, March). Improving the Robustness to Targeted Attacks in Software Defined Networks (SDN). In DRCN 2017-Design of Reliable Communication Networks; 13th International Conference; Proceedings of (pp. 1-8). VDE.

[42] Girisankar, S. T., Truong-Huu, T., & Gurusamy, M. (2017, January). SDN-based dynamic flow scheduling in optical data centers. In Communication Systems and Networks (COMSNETS), 2017 9th International Conference on (pp. 190-197). IEEE.

[43] Malik, A., Aziz, B., Adda, M., & Ke, C. H. (2017). Optimisation methods for fast restoration of software-defined networks. IEEE Access, 5, 16111-16123.

[44] Sminesh, C. N., Kanaga, E. G. M., & Ranjitha, K. (2017, January). Flow monitoring scheme for reducing congestion and packet loss in software defined networks. In Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on (pp. 1-5). IEEE.

[45] Schnepf, N., Badonnel, R., Lahmadi, A., & Merz, S. (2017, July). Automated verification of security chains in software-defined networks with synaptic. In Network Softwarization (NetSoft), 2017 IEEE Conference on (pp. 1-9). IEEE.

[46] Tangari, G., Tuncer, D., Charalambides, M., & Pavlou, G. (2017, May). Decentralized monitoring for large-scale Software-Defined Networks. In Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on (pp. 289-297). IEEE.

[47] Chen, G. R., Li, C. Y., & Wang, K. (2017, May). ReSDN: A lightweight solution for data-plane state recovery in software-defined networks. In Communications (ICC), 2017 IEEE International Conference on (pp. 1-7). IEEE.

[48] Peizhe, L., Muqing, W., Wenxing, L., & Min, Z. (2017). A game-theoretic and energy-efficient algorithm in an improved software-defined wireless sensor network. IEEE Access, 5, 13430-13445.

[49] Boite, J., Nardin, P. A., Rebecchi, F., Bouet, M., & Conan, V. (2017, July). Statesec: Stateful monitoring for DDoS protection in software defined networks. In Network Softwarization (NetSoft), 2017 IEEE Conference on (pp. 1-9). IEEE.

[50] Song, L., Fragouli, C., & Zhao, T. (2017, June). A pliable index coding approach to data shuffling. In Information Theory (ISIT), 2017 IEEE International Symposium on (pp. 2558-2562). IEEE.

[51] Alharbi, T., Layeghy, S., & Portmann, M. (2017, November). Experimental evaluation of the impact of DoS attacks in SDN. In Telecommunication Networks and Applications Conference (ITNAC), 2017 27th International (pp. 1-6). IEEE.

[52] Wang, T., Xu, H., & Liu, F. (2017, June). Multi-resource load balancing for virtual network functions. In Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on (pp. 1322-1332). IEEE.

[53] Durner, R., Lorenz, C., Wiedemann, M., & Kellerer, W. (2017, July). Detecting and mitigating denial of service attacks against the data plane in software defined networks. In Network Softwarization (NetSoft), 2017 IEEE Conference on(pp. 1-6). IEEE.

[54] Suh, D., & Pack, S. (2018). Low-Complexity Master Controller Assignment in Distributed SDN Controller Environments. IEEE Communications Letters, 22(3), 490-493.